

# Java in Two Semesters: Featuring JavaFX (4<sup>th</sup> Edition)

Charatan and Kans

## Installing and Using NetBeans™

### 1. Installing NetBeans™

Before downloading and installing the NetBeans™ tool ensure that you have downloaded and installed the Java SE Development Kit (JDK) from the Oracle™ site. To follow the examples in the text it is recommended that you download JDK version 8:

<https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

Once you have downloaded and installed the JDK you can then download and install the Apache NetBeans™ IDE from the following site

<https://netbeans.apache.org/download/nb100/index.html>

The NetBeans™ tool is downloaded as a zip file. Unzip this file and locate the NetBeans™ executable file in the **bin** folder.

### 2. Using NetBeans™

#### 2.1 Configuring NetBeans™

NetBeans™ comes preconfigured with various settings that can be customised if required. For example, to follow the curly bracket style of the text (with each bracket on a new line) follow these instructions:

- Navigate to **Tools -> Options -> Editor**
- Navigate to **Editor -> Formatting** and set the **Braces** options as shown in figure 1

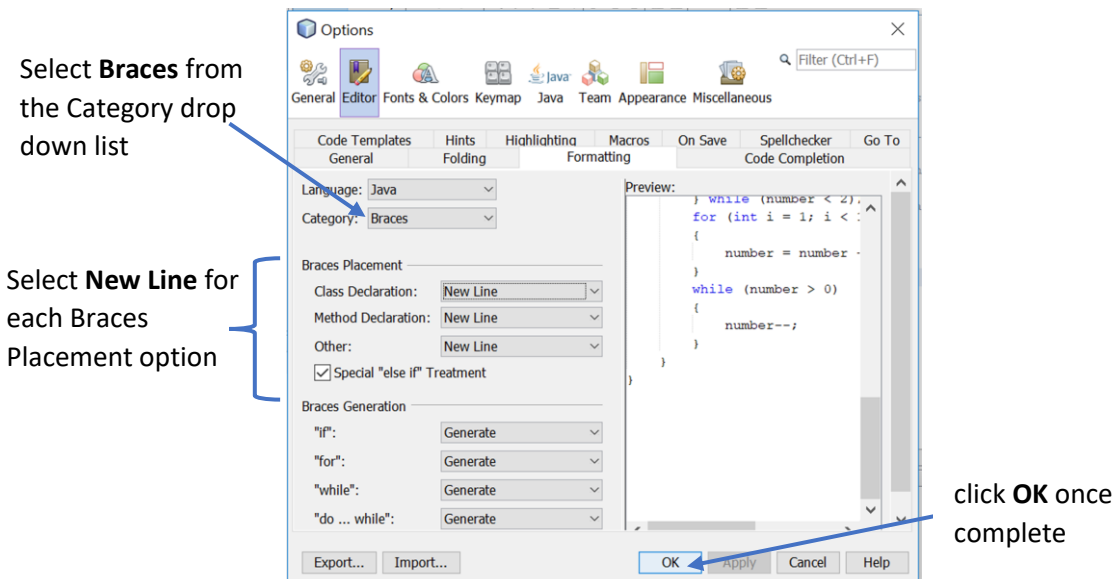


Figure 1

## 2.2 Starting a new project

All the files associated with an application should be kept in a single project. It is recommended that you create a separate project for each application that you are creating. To create a project, choose *New Project* from the file menu as shown in figure 2.

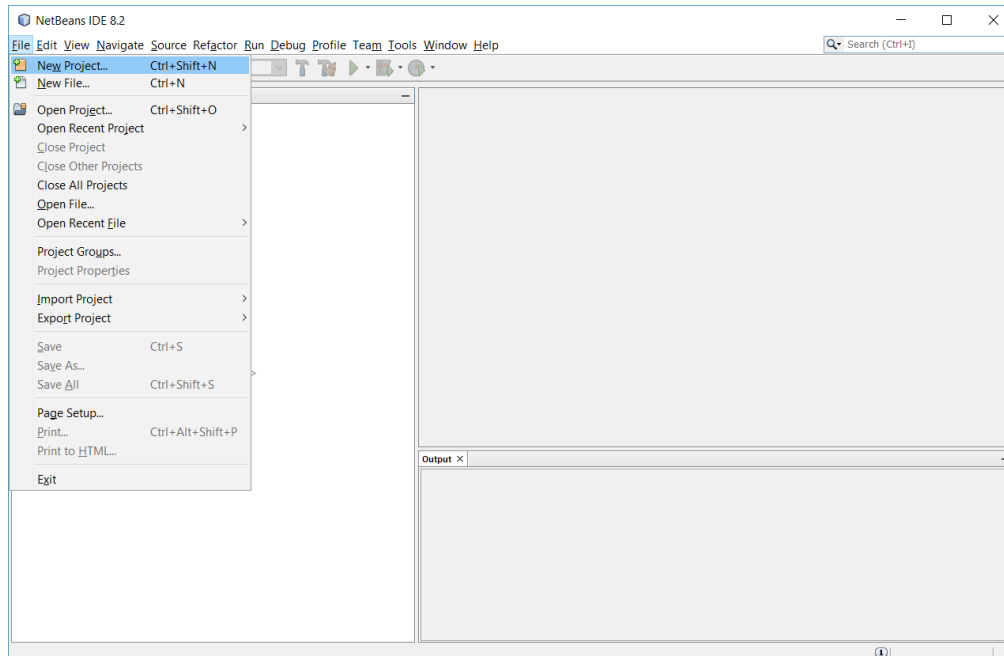


Figure 2

You will now see the screen shown in figure 3. From the *Categories* window, choose Java, and then choose *Java Application* from the *Projects* window. Then press *Next*. Leave the *Filter* field blank.

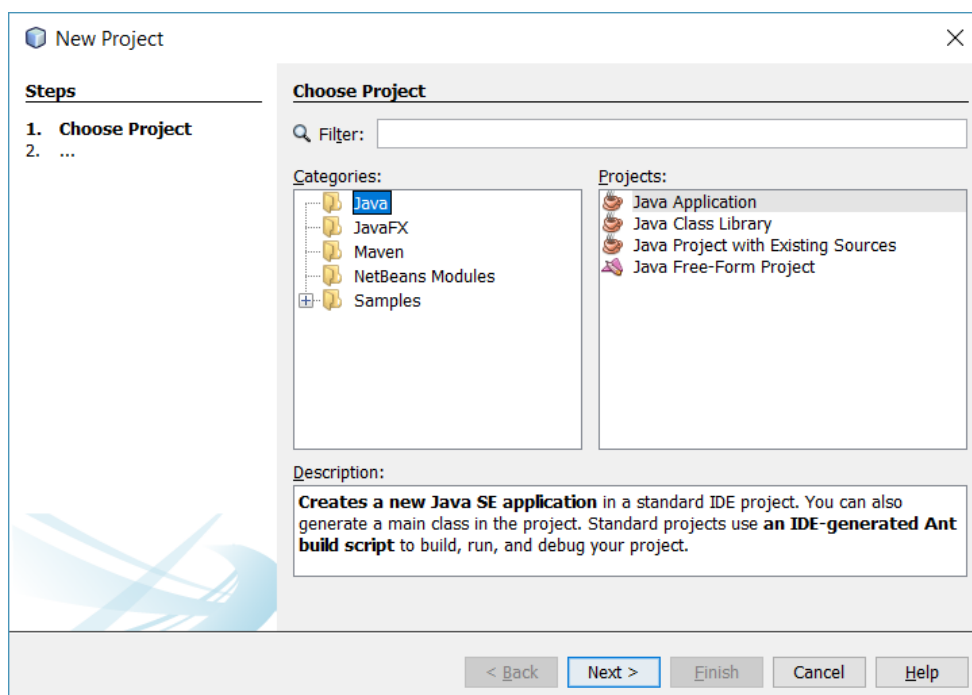


Figure 3

You will now see the screen shown in figure 4.

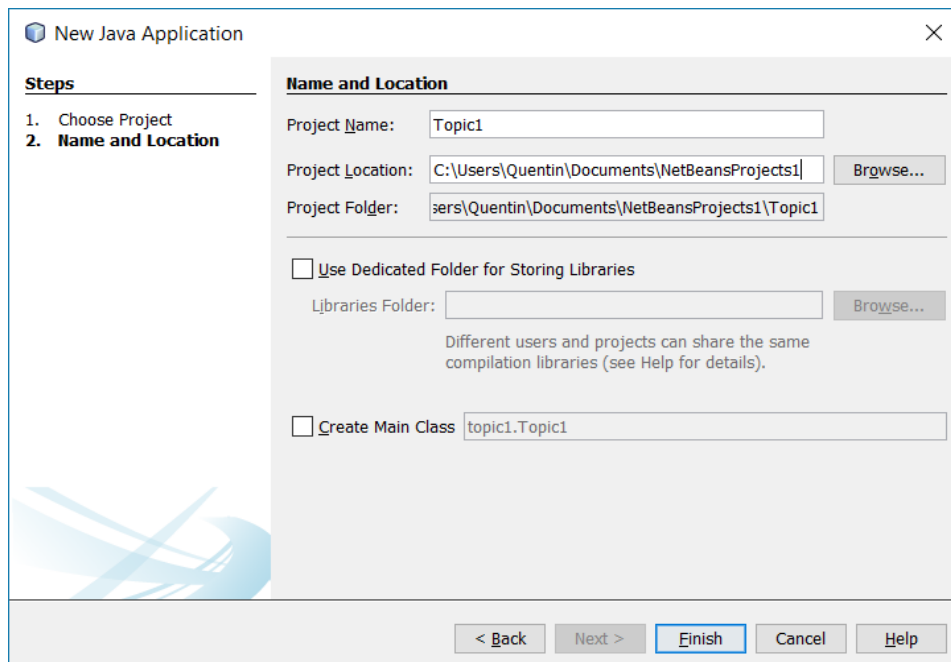


Figure 4

Enter the name of the project (in this example *Topic1*) in the *Project Name* field.

In your early programs you are not going to create packages for your applications, so you want all your source files to be kept in what NetBeans™ calls a **default package**. The easiest way to achieve this is to uncheck the *Create Main Class* box, and create your main class later. However, if you do want to create the main class at this stage (and you want it to reside in the default package), change its name to include only the class name – so in figure 4, change *topic1.Topic1* to *Topic1* – or give it the name you require such as *HelloWorld*.

Now you can press *Finish*. Your screen will now appear as in figure 5, with *Topic1* listed as your only project.

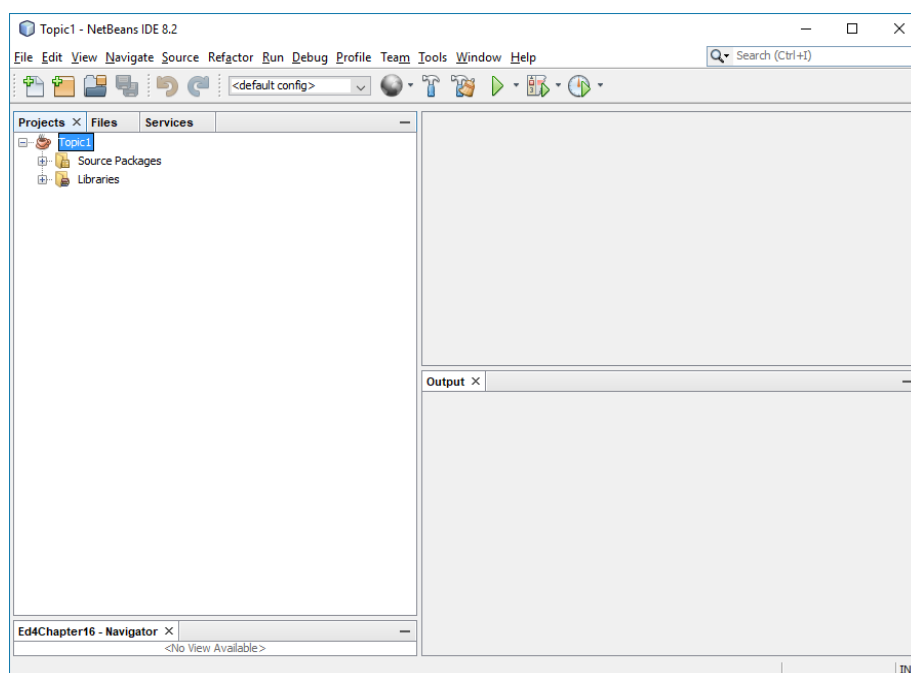


Figure 5

## 2.3 Adding new files to the project

To add a new file to your project, you can either choose *New File* from the *File* menu, press the + sign in the top left hand corner, or type Ctrl+N. You will see the screen shown in figure 6.

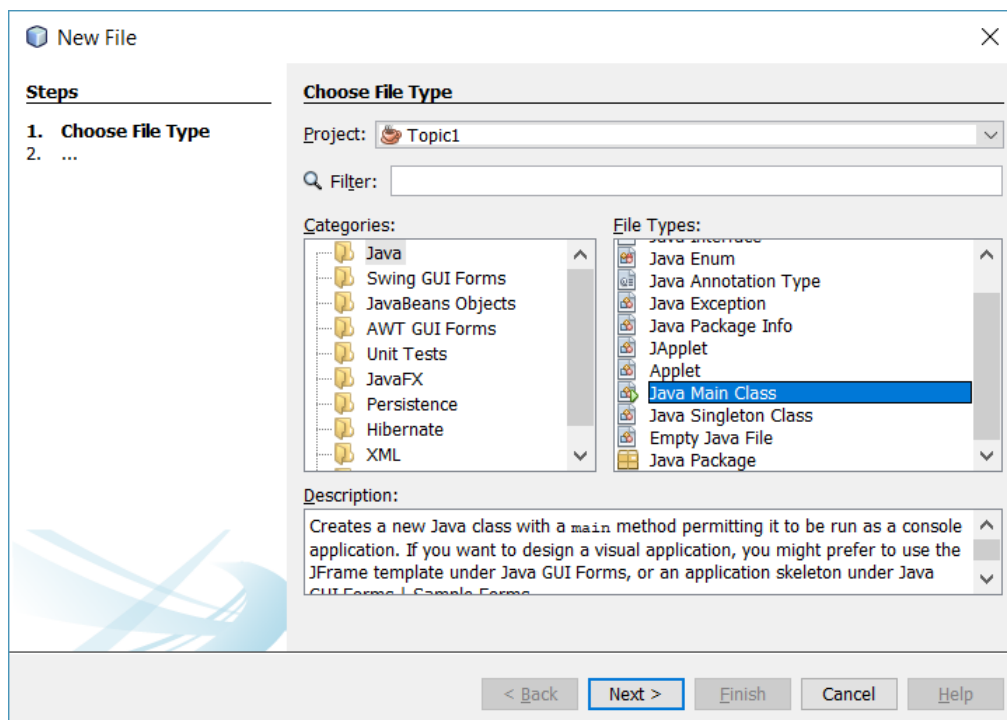


Figure 6

Choose *Java* from the categories section, and from the *File Types* window choose either *Java Main Class* or *Empty Java File*, depending on which you want.

Pressing *Next* will cause a new screen to appear, as in figure 7.

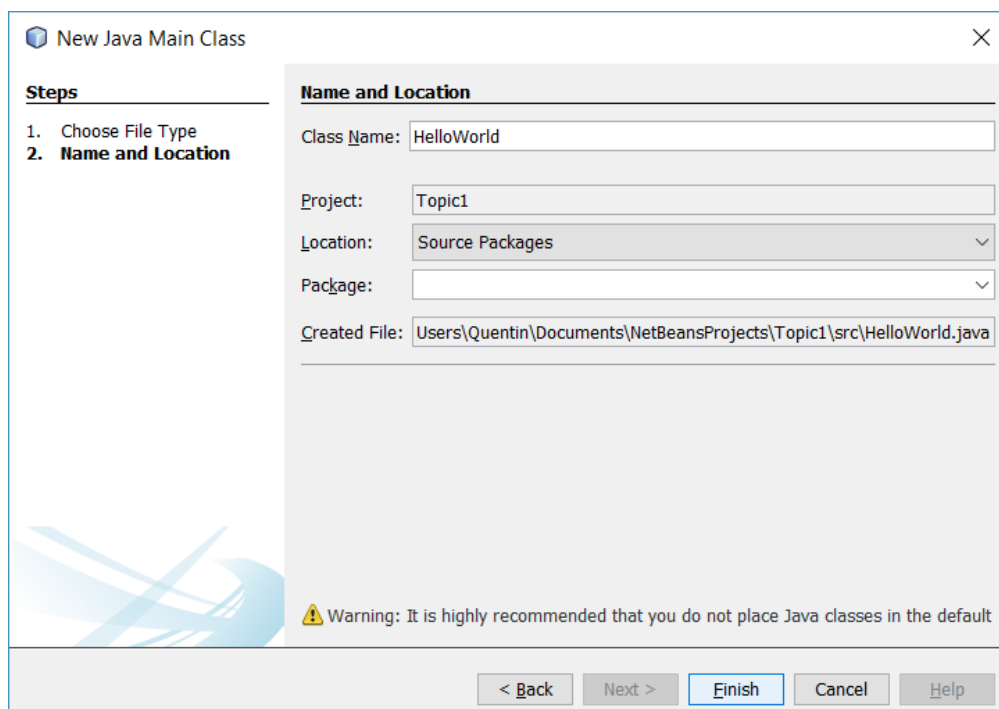


Figure 7

Type in the name that wish to give to your class (*HelloWorld* in this example), and press *Finish*. The class will be listed in the default package, and the code will appear in the tab as shown in figure 8. If you had chosen an empty class, this would be blank.

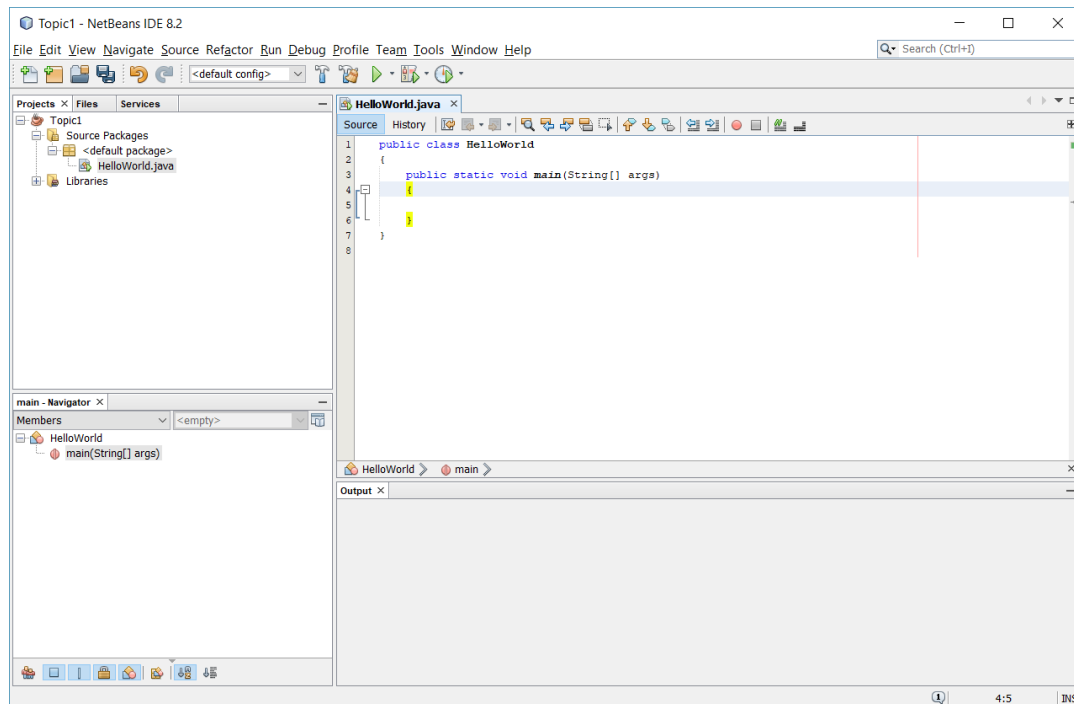


Figure 8

## 2.4 Compiling and running applications

Classes are automatically compiled as soon as they are written. Error messages are indicated in red and are revealed when you hover over the red icon with the mouse (as shown in figure 9).

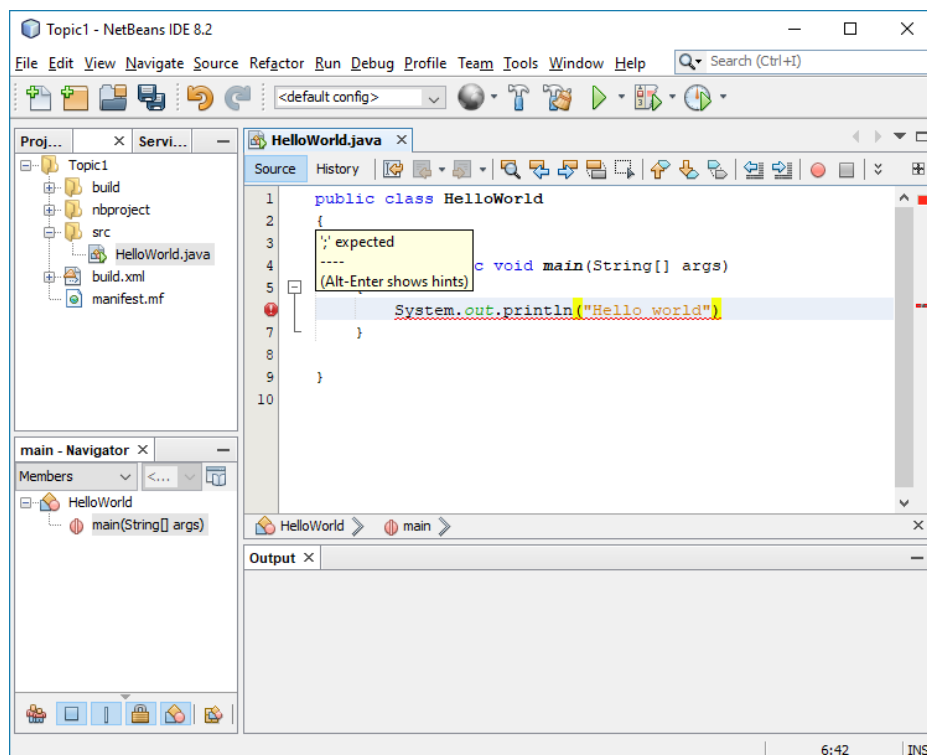


Figure 9

Programs can be run by pressing shift+F6 on the active window containing the main class; alternatively you can right-click on the class in the list and choose *Run File* from the drop down menu (or again press shift+F6).

Running a non-graphics program will cause the output to be displayed in an output window as shown in figure 10. This window can be made to float, if you prefer (this is most easily done simply by dragging it).

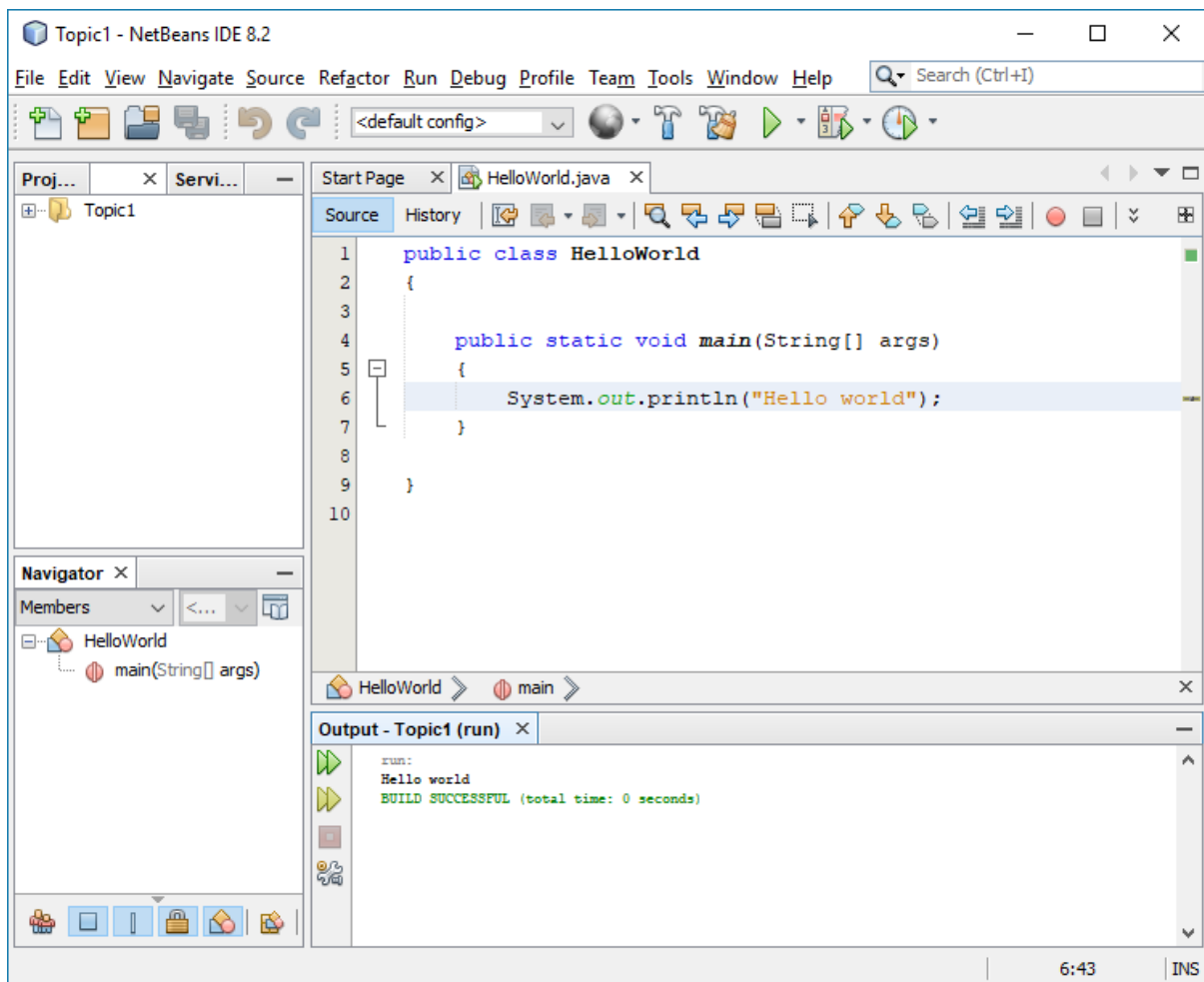


Figure 10

## 2.5 Adding existing classes to projects

Existing classes (that is to say, the source code) can be added to the project simply by placing them in the correct folder.

For example, if the default directory for your NetBeans projects is called *NetBeansProjects*, and your project is called *Topic1*, then the *.java* files are kept in a folder called *src*, as shown in figure 11. If you had created your own package, then files residing in that package would appear in a subfolder of *src* with the package name.

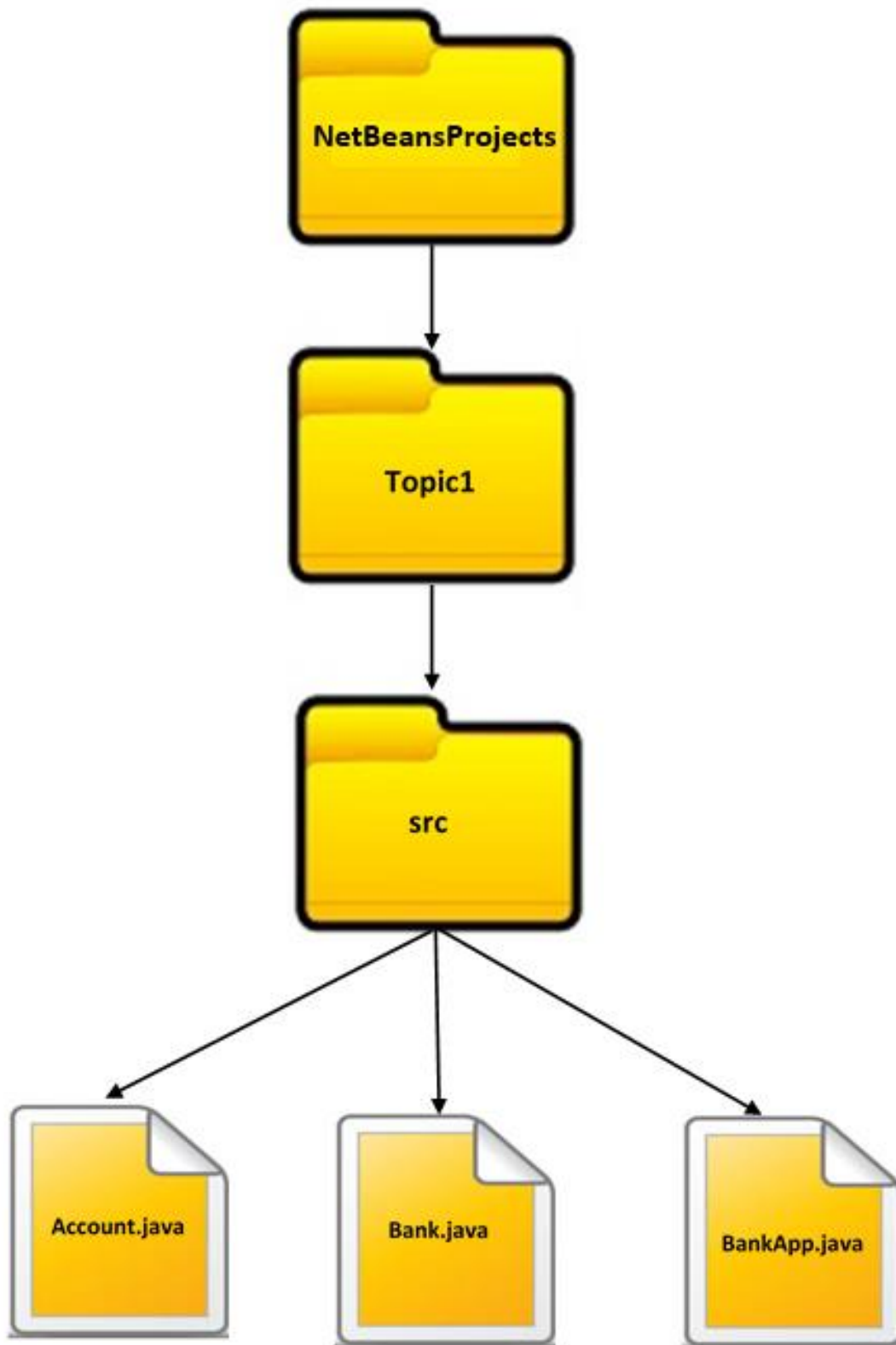


Figure 11

### 3. Creating *Jar* files with NetBeans™

#### 3.1 Creating an application

All the files associated with your application should be contained within a single project. In the example shown in figure 12, a project called *CaseStudy1* has been created.

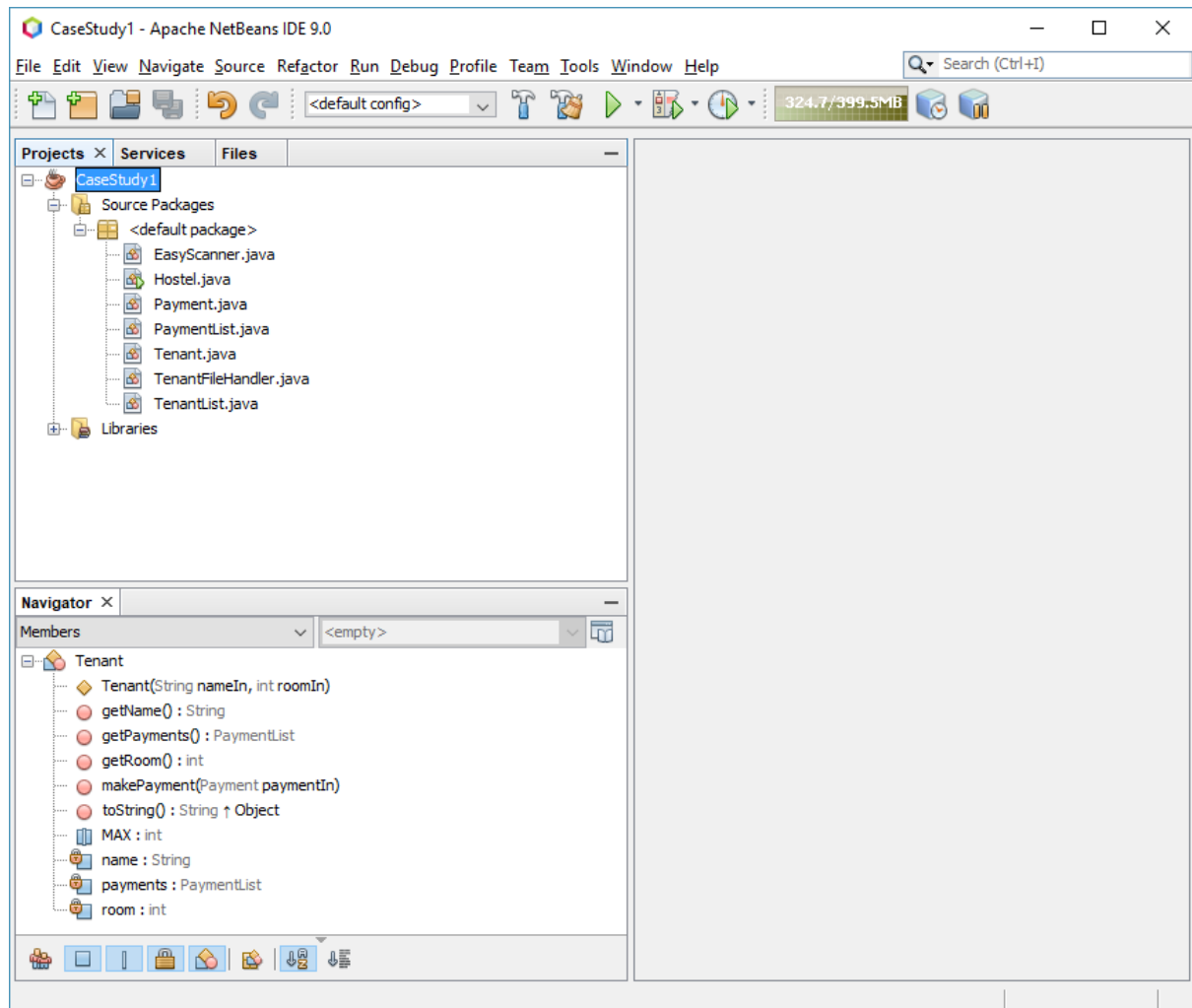
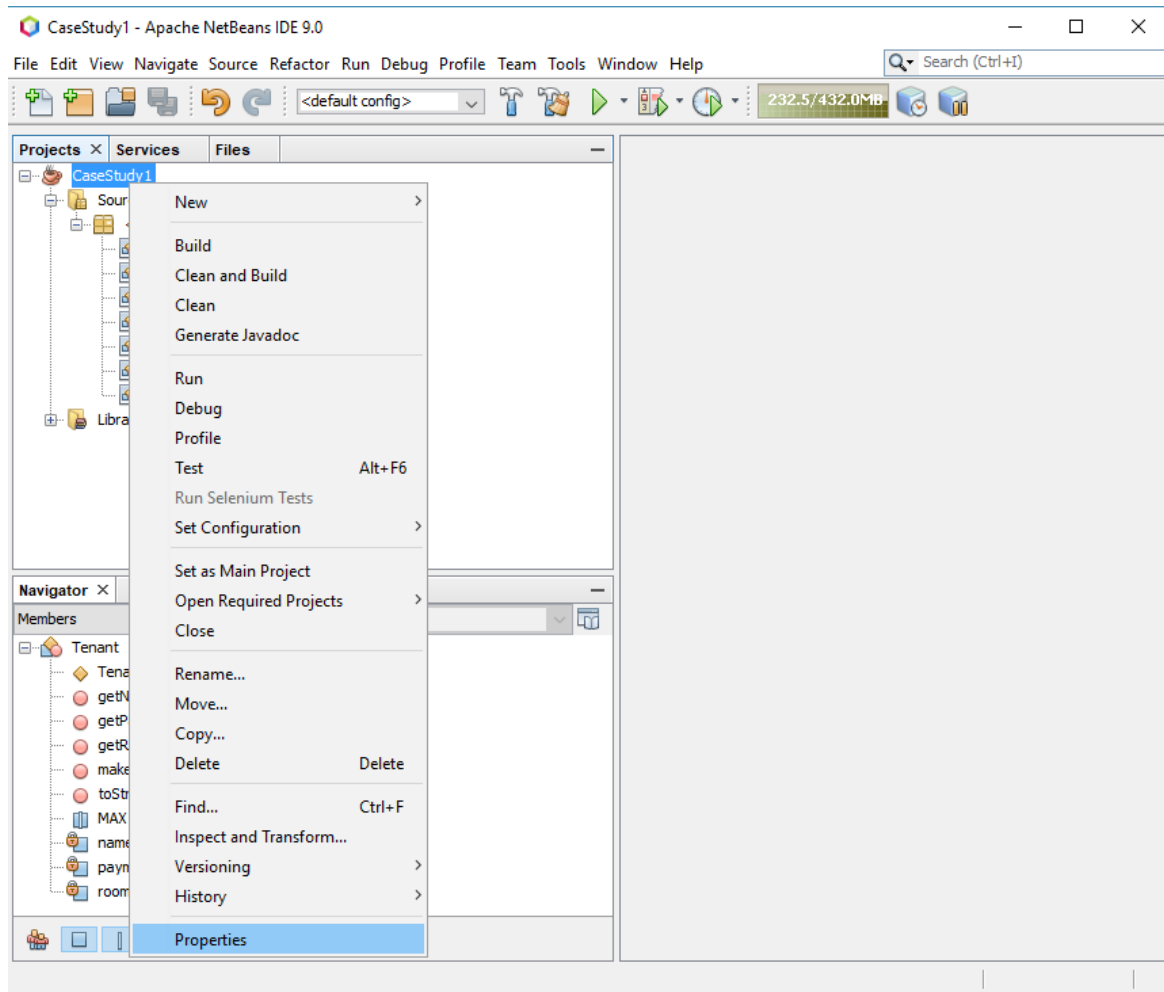


Figure 12

#### 3.2 Setting the project properties

Right-click on the name of the project, as shown in figure 13, then select *Properties*.

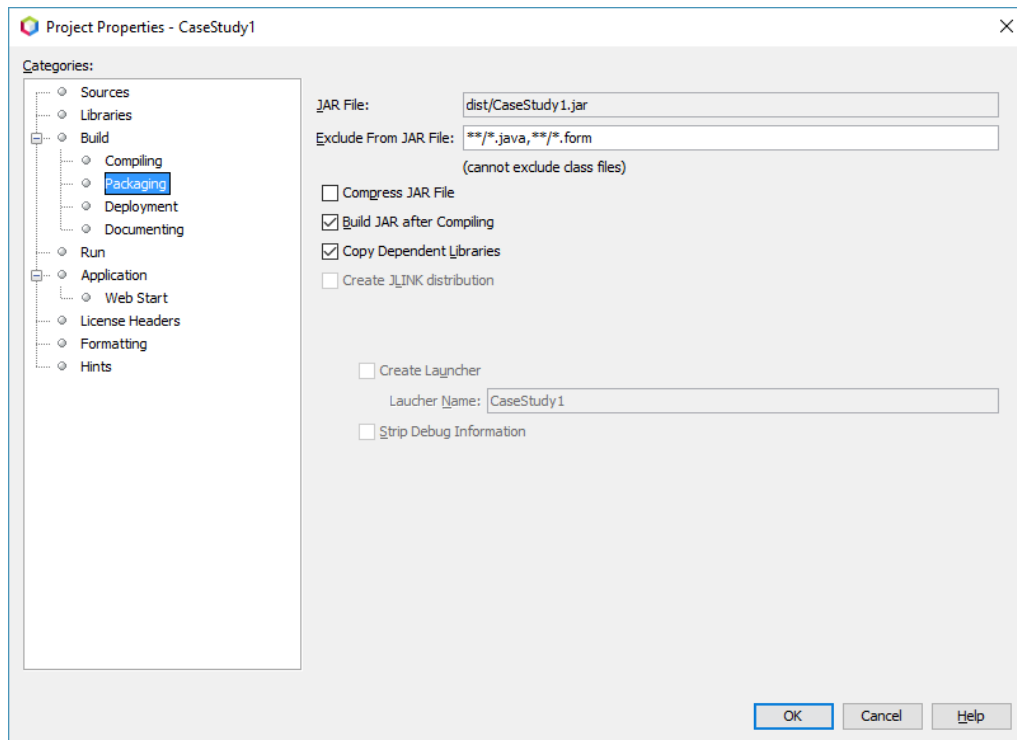




**Figure 13**

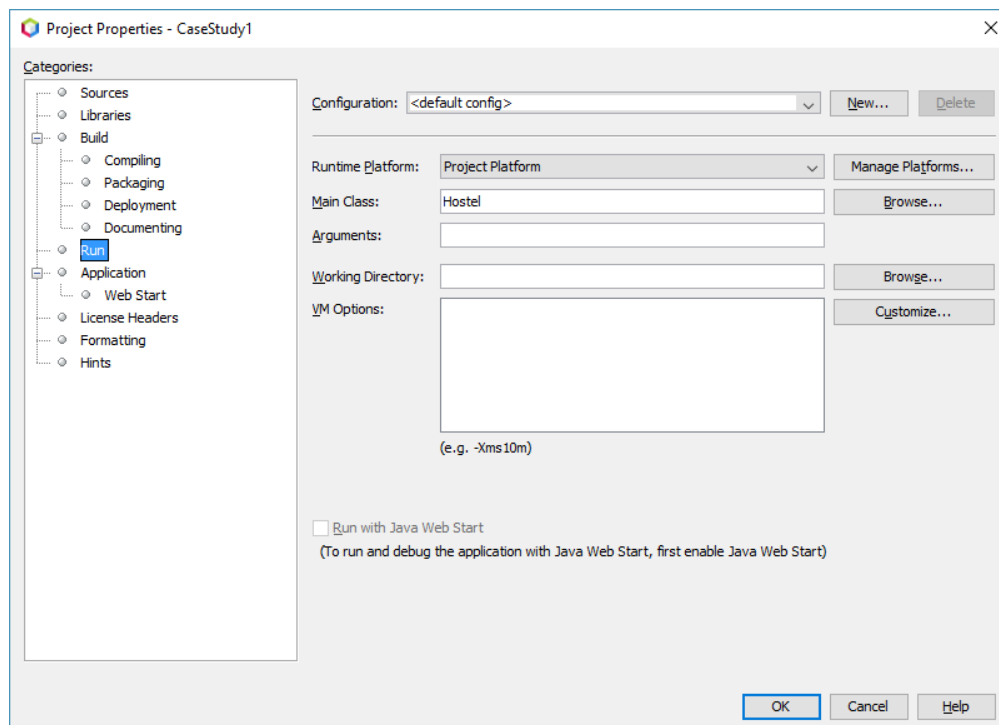
The screen shown in figure 14 will appear.

Highlight *Packaging* and make sure that *Build JAR after Compiling* is ticked.



**Figure 14**

On the same screen choose *Run* (figure 15). Then, using the *Browse* button, select the main class (*Hostel* in the example shown).



**Figure 15**

### 3.3 Creating the *Jar* file

Right-click on the name of the project again and then select *Build*, as shown in figure 16 (you could also choose *Clean and Build*, which will erase any previous files). Your *Jar* file will be created and will be saved in a directory called *dist*, in the project folder. It will have the same name as the project (in this case it will be called *CaseStudy1.jar*). You can change the name if you wish (but you must leave the *.jar* extension).

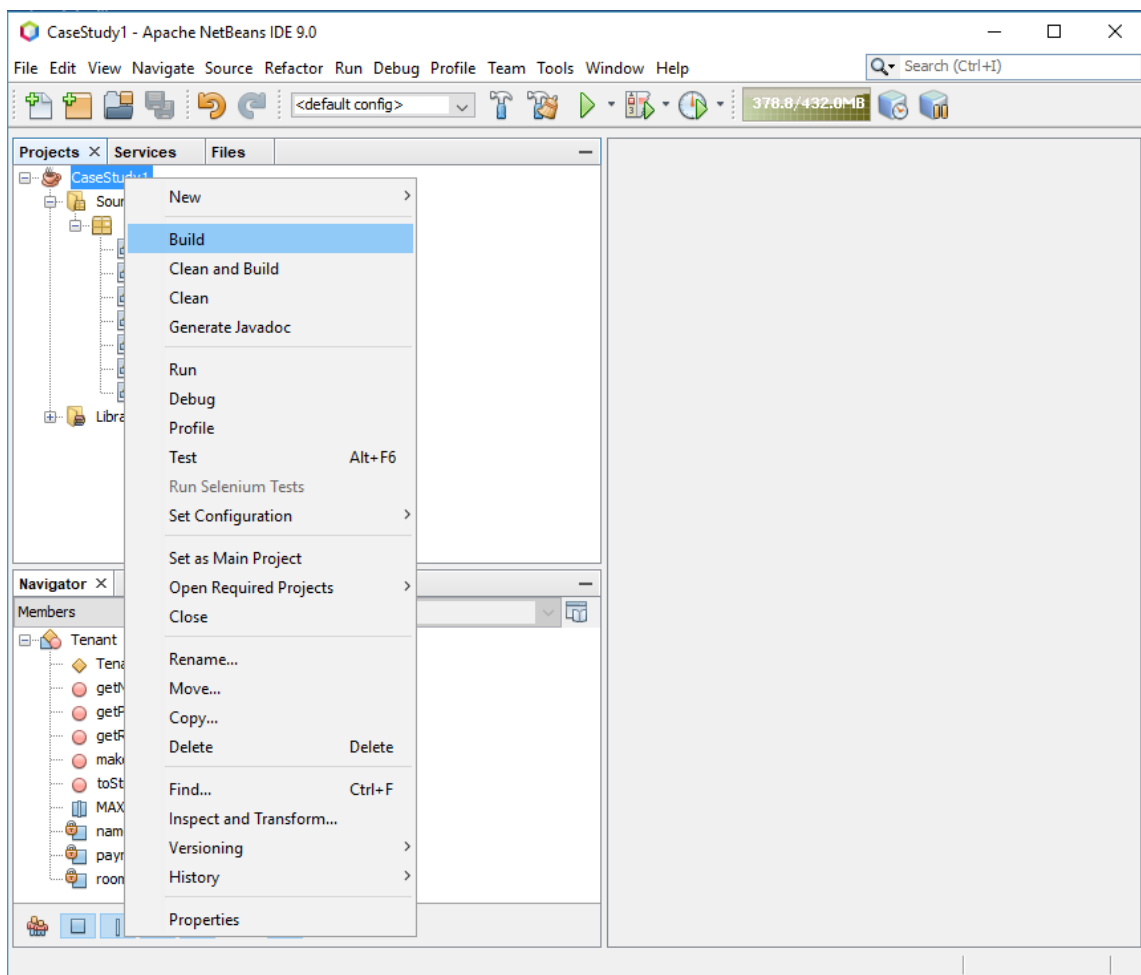


Figure 16

Java in Two Semesters

Featuring JavaFX

Charatan, Q.; Kans, A.

2019, XVIII, 716 p. 1251 illus., 100 illus. in color.,

Hardcover

ISBN: 978-3-319-99419-2